admins may grant the "Locals" group "enabled" permission over the BookMark module. This permission ONLY applies within SiteB.

System Admin Alex requires Site B to use the MyMenu. The menu will be part of Site B's navigation tree, and Site B cannot remove it. If the Engineering group is allowed to logon to Site B, they will see the menu item "Transportation Options" because they were granted permission to see it in the system context. In addition, Site B Admins may grant the "Locals" group permission to see the menu item in the context of the site.

The act of sharing does not change existing permissions. When a component moves into a site, the site admin can control which groups may see it within their site

2.2.9 Moving Components with End User Permissions between Sites

Let's throw in SiteC to muddy the waters. In this site is the Engineering group, Local group, and Bookworm group. Also, SiteC has been shared (and is using) MyMenu and BookMark. SiteC has not granted any end user permissions.
When the Bookworm group comes to the site, they will be able to see the BookMark module, because they were granted permission in the system context. They will not see the Transportation Options page because they have not been granted permissions to see the link to Transportations Options in any context. When the Engineering group comes to the site, they will see the Transportations Option page because they were granted permission to view the link. They will not see the BookMark page. The "Locals" group will not see either the BookMark or the menu item page Transportation Options. The permissions granted in the Site B context do not apply when they are visiting Site C.

2.2.10 The Server Admin on an End User Site

The server admin can always logon to any end user site. Once they are logged on, they receive all site administration privileges. This means that, once in a site, they always get the "go to admin console for this site" link.

For end user site viewing permissions they are stripped of their server admin status. Their permissions are determined by membership in all their other groups. Often, this will mean that they aren't in any group that the site has explicitly permissioned except for the Registered Users group.

3 Import /Export of components

3.1 Overview

A web portal is composed of many "components". Examples of components are:

- The modular applications that run within the portal (called "modules" or "portlets")
- The elements that control the look, feel, and navigation of the portal (called "templates" and "styles")
- Some advanced portal systems, instead of running one portal, run multiple portals, also called "sites". These "site" components contain other components, such as modules and styles.

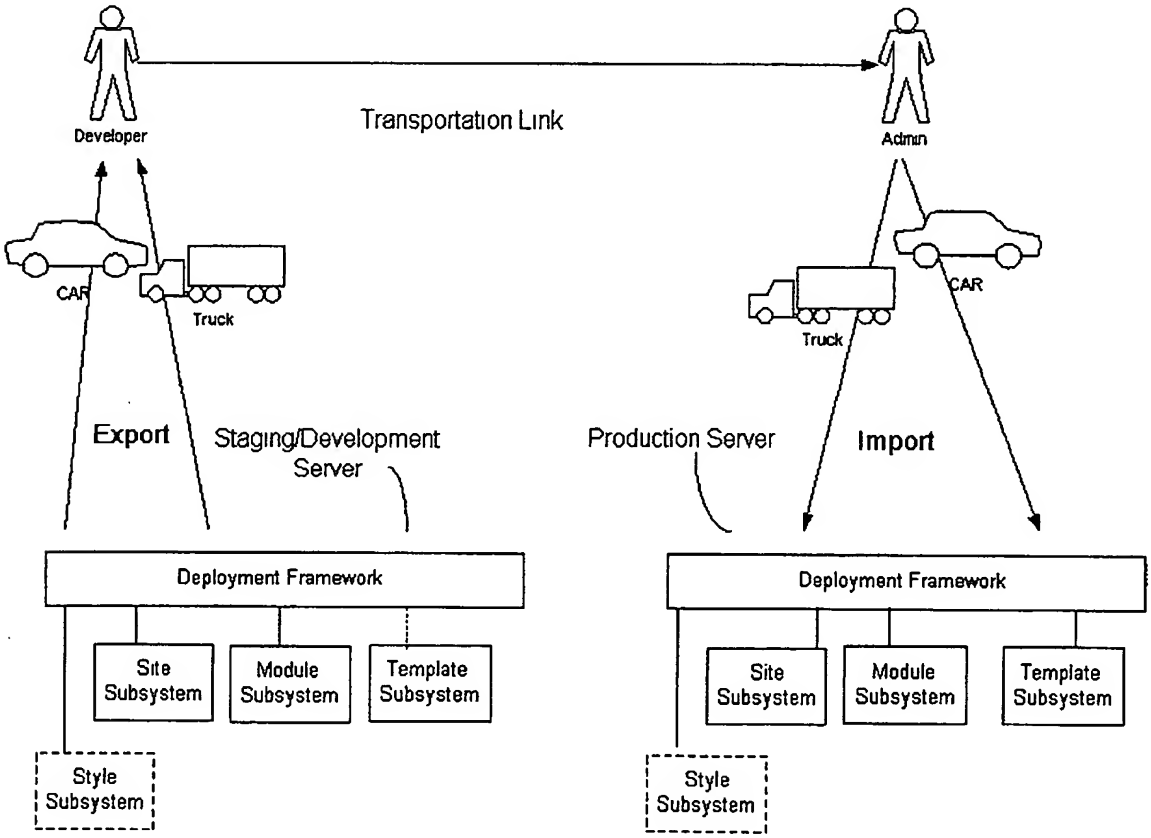
There exists a need to automate the export and importation of components from one portal to another, through a simple graphical interface. Uses of this include, but are not restricted to:

1) Moving components from a staging or development portal environment to a production system
2) Distributing components to all machines a global network of portal systems, for scalability and bringing servers closer to the people that use them
3) Moving components between two independent portals that happen to share some functionality
4) Allowing the packaging and commercial sale of these components to portals .

In the prior art many methods were used to accomplish the migration of code from the staging server to a production server. One of the prior art methods used was to physically move the files by individual manual selection or using a web application archive (WAR) file deployment. There are many drawbacks to the prior art. Selecting loose files is both cumbersome and extremely prone to error. Secondly, there is no means in the prior art to export/import components that have both file assets along with non-file assets such as database objects (permissions, user preferences, etc) and instantiated runtime objects (as opposed to static classes). In prior art, these non-file assets would then have to be manually recreated at the production server. The present invention accomplishes the above prior art drawbacks as illustrated below.

3.2 Logical Diagram

This logical diagram illustrates how customers will move from development/staging to production.



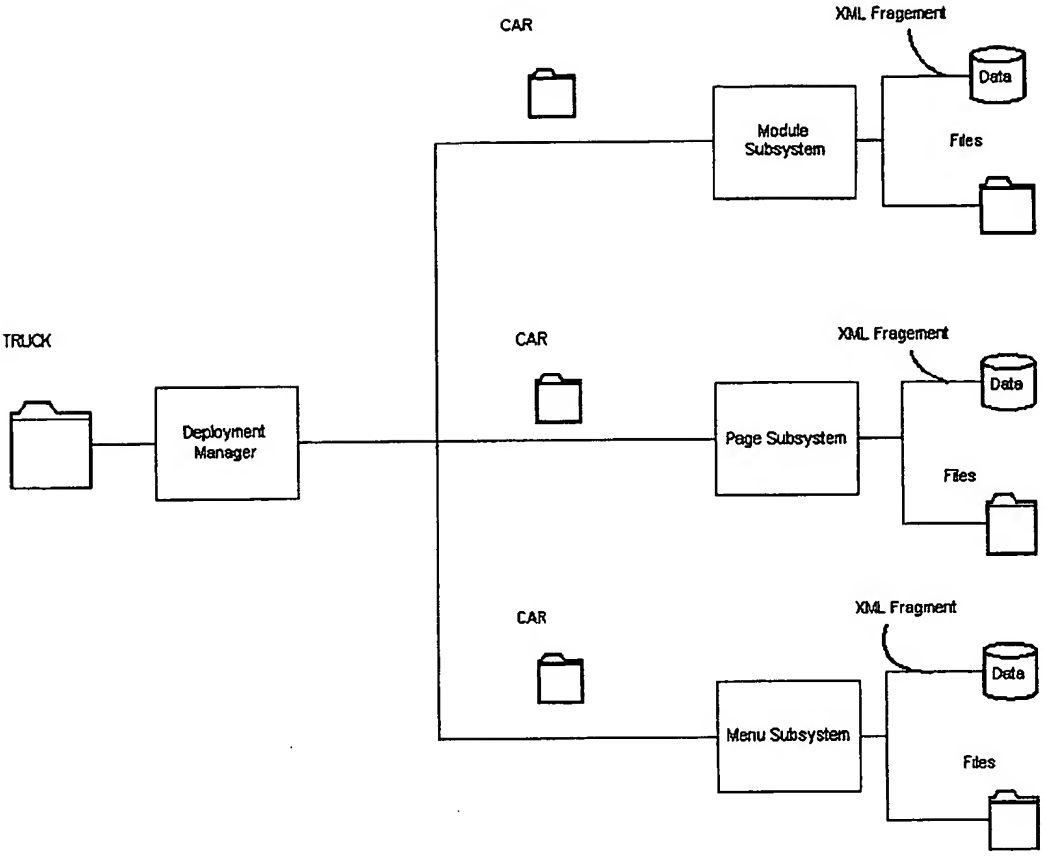
3.3 Deployment Manager/Export

Components consist of file and non-file assets: file assets include resources such as code (JSP or ASP pages, Java or other programming language classes) and images (GIFs, etc) and non file assets ("data assets") include in-memory instantiated programming language objects (as opposed to static class files) permissions, user preferences, settings, menus, user groups etc.

When an administrator for the site or server exports a site through the administrator graphical user interface (Admin GUI), a deployment manager API is implemented. The deployment manager acts much like a dependency sniffer, which queries every subsystem associated with the selected site or components.

The deployment manager will query each component subsystem for all associations with the site or component to be exported. The deployment manager for example will query the module subsystem for any modules that have been associated with the particular site or component to be exported. The module subsystem will then collect all the file assets (jsp files, images etc.), and the non-file database assets (permissions, settings, etc) for the module associated to site or component. The non-file asset data will be constructed as an XML fragment within a predetermined schema. The XML fragment also contains the descriptor ID's for each component to be added to the CAR file and the ID of the subsystem that component originated from. The two sets of data from every subsystem will then be archived together in a component archive file (CAR). Each CAR file is collected and archived again into a larger truck file, which is then ready for transport to a designated server in a zip format. This file is downloaded to the machine that the request (export request) is made from via an ftp like process. It is important to note that every component for in a given subsystem maybe collected as a group or individually depending on the permission setting of the administrator or what components the administrator is attempting to transport to another server.

A Site CAR will include all Page, Menus, Module Instances, User Groups and Themes, along with site configurations. When the Site is delivered to production, the site will be reconstructed, with all appropriate permissions and settings, and shares reestablished. The below figure illustrates a subset of subsystems that are queried by the deployment manager for the export of a selected site. Multiple sites maybe selected at one time, which in turn produces a Truck file of all the collects CARS from each selected site.



3.4 Deployment Manager/Import

When an administrator for the production server is ready to "import" the TRUCK/CAR that was downloaded to the production server. It should be noted that the IMPORT function of the deployment manager works almost in reverse of the EXPORT function that is illustrated in the above figure. The administrator selects the import link from the administrative GUI and selects the appropriate file to be "uploaded". The deployment manager then takes the relevant CAR files and extracts them. The deployment manager then examines the XML files for each extracted CAR and determines by the descriptor ID's, what the extracted component is and which subsystem that component belongs. The subsystem then takes the particular component files, extracts them to the appropriate location in the file system. The XML fragment is then parsed and the objects contained therein are instantiated in the database or other relevant location. The deployment manager eliminates the need for the administrator to hand deliver files to the correct location on the file system and to reset all of the permissions, settings, etc. that was done on the staging server. Plus, the production system does not need to be restarted, therefore resulting in a significant lose of downtime for the server..

3.5 Import/Export of Shared Components

When a site is exported, it takes with it all objects that have been shared to it. If donor sites do not exist on the target installation, these objects revert to System ownership. Even when the donor site is imported, these objects will still have system ownership. Therefore, admins should import and export sites that are net donors first, and then net recipients.

3.6 Versioning

The system will support both major and minor version numbers. However, each version can only have one build. For example, the software residing on the server will support having minor version 1.2 build 5, minor version 1.3 build 6 and major version 2.0 build 1. But if the user tries to upload 2.0 build 2, it will overwrite the existing 2.0. In addition, if they have 3.4 build 1 and reimport 3.4 build 1, the existing CAR will be replaced with the incoming CAR, and the site will be overwritten with the newer CAR.The deployment framework will keep all CAR versions that are imported into the system. When a new site CAR is uploaded to the system, the system will deploy the site CAR only if it is

a) the same version (major and minor) and build number as the current site or

b) a larger version or build number than the current site.

The largest version number will always be the deployed site. In addition, when a site is deleted, all versions of the site CAR should also be deleted from the system. The version number will not be exposed or manipulated through the User Interface. When an administrator exports a site, if there is no version number, the system automatically assigns it a version number of 1.0

4 Glossary

4.1 Site

A site is considered to be a collection of administrable objects (modules, pages, users, etc) given a single identity by a shared look-and-feel, a shared set of navigation links, and an administrative group that automatically receives permissions to administer all or nearly all of the objects within the site.

4.2 Referring Site

This is the site that is the referrer to the admin site. A admin user must first choose a site to login to, and then click through to the admin site via the dynamic "Administration" link.

4.3 Site Admins (members of the Site Administration group)

Site admins are members of the Site Administration group for a site. A site administration group is a usergroup that is automatically created whenever a user creates a new site. Site admin groups are inextricably linked to sites; a site admin group will exist as long as its site exists.

Site admins are like server admins for a single site. Members of a site admin group for a site will be the most powerful administrators of that site. Less powerful site administrators should be placed into other groups that have limited delegated administrative privliges on a site – they should not be placed in the site admin group.

4.4 Server Admins

Server admins are installation-level administrators who have complete privileges over all aspects of an installation. The server admin setting is a per-user setting that entirely bypasses all system permissions, effectively giving server admins all permissions in the system.

4.5 The Everyone Group

This is a special conceptual group which allows batch permissions to be set for all groups. It isn't an actual group but rather a flag that lives on every permission which short-circuits the group-based permissions lookup.

4.6 Site Repository

Every site in an installation has a single site repository. The site repository is used to store all items (modules, templates, pages, etc.) that are available to be used on a site. Administrators of a particular site will, by default, have permission to view and modify the contents of their site's site repository. The act of adding an item into a site's repository automatically causes that object to be permissioned so that the site's site admins will have permission to administer the object.

In addition to providing for automated permissioning of these objects, the site repository also provides for site-based grouping, or partitioning. The site repository can constrain the views of site administrators to only those objects that the repository contains. This provides a security model for delegated site admins that enables different site admin groups to potentially use completely different sets of objects for building and administrating their sites.

4.7 Site Context

An delegated admin will be able to perform various actions in one of three contexts: in the context of a particular site, in the context of the shared repository, or in the system context.

If a user is working in the context of a particular site, then the actions that he or she takes when in that context will pertain to that site. For example, if a user is working within the context of SiteX, and that user creates a module, then the module will automatically go into the SiteX repository.

4.8 Shared Repository

The *shared repository* is a collection of items in the system that are available to be used on multiple sites. A single item can be placed in any number of repositories (site, shared, or system), or in no repositories at all. All the objects that can be placed in a site repository (listed above) can also be placed in the shared repository for the purposes of sharing.

4.9 System Repository

The *system repository* is the system-wide view of all items that a user has permission to see, regardless of what repositories those items are in.

4.10 System Context

Uber and delegated admins will be able to work in a context called the system context. In this context, the server admin will have access to all objects in the system, and delegated admins will have access to all objects in the system that they have permission on.

4.11 The XML-based Sites Utility

Administrates sites outside the limitations of the web-based admin UI with a flexible, powerful, XML-based sites utility. Copies sites from one 4.0 installation to another. Also can be used to archive sites to disk.

3.12 Deployment Framework

This is the software framework that manages and records the deployment state of components in the system, throughout the deployed cluster.

3.13 CAR

Zip Format File. Component Archive. Contains resources which can be deployed

3.14 TRUCK

A CAR file that contains other CAR files.